

LISTING OF CLAIMS

1. (Currently Amended) A method for testing software generating a map that associates a graphics element of a graphical user interface of a software application with an executable feature of the software application, the method comprising:

6 retrieving information descriptive of a the graphics element rendered

7 during execution of the software being tested, the information

8 identifying including an executable feature associated with the graphics element;

9 storing an association between the executable feature and the graphics element in a map data structure, wherein the map data structure is accessible by an application driver for driving the software application; and

10 11 executing the an executable feature stored in association with thea graphics element; and

12 13 updating the association in the map data structure upon execution of the executable feature.

15 16 2. (Currently Amended) The method of claim 1 further comprising, in

17 response to executing the executable feature:

18 displaying exposing a second graphics element;[:]

19 retrieving information descriptive of the second graphics element, the information including a second executable feature associated with the second graphics element;

20 21 storing the second executable feature in association with the second graphics element in the map data structure; and

22 23 executing the second executable feature stored in association with the second graphics element.

3. (Original) The method of claim 1 wherein the retrieving comprises capturing information pertaining to the graphics element.

4. (Currently Amended) The method of claim 1 wherein the storing includes updating an indicator associated with the graphics element when the an executable feature stored in association with the graphics element is executed.

5. (Currently Amended) The method of claim 1 wherein the storing includes organizing the retrieved information such that the an executable feature stored in association with the graphics element can be interpreted by a computer-executable application capable of accessing the retrieved information.

6. (Currently Amended) The method of claim 1 wherein the storing includes organizing the retrieved information such that the an executable feature stored in association with the graphics element can be interpreted by a user capable of accessing the retrieved information from memory.

7. (Currently Amended) The method of claim 1 further comprising selecting the executable feature based on the association stored in the map data structure wherein the executing comprises selecting from the stored information an executable feature stored in association with a graphics element.

8. (Original) The method of claim 7 wherein the selecting comprises selecting an executable feature not previously executed.

9. (Original) The method of claim 8 wherein the selecting comprises reviewing an indicator to select an executable feature not previously executed.

1 10. (Original) The method of claim 7 wherein the selecting comprises
2 selecting executable features in a depth-first mode of operation.

3 11. (Original) The method of claim 7 wherein the selecting comprises
4 selecting executable features in a breadth-first mode of operation.

5
6 12. (Canceled)

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

1 13. (Currently Amended) A system for generating a map that associates a
2 graphics element of a graphical user interface of a software application with an
3 executable feature of the software application, the system comprising:

4 a capture agent for retrieving information descriptive of the a plurality of
5 graphics elements rendered during execution of the software application, the
6 information including an executable feature associated with each~~the~~ graphics
7 element;

8 an application driver for storing an association between each~~the~~ executable
9 feature and the corresponding graphics element in a map data
10 structure and for deterministically selecting one of the executable features
11 stored in the map data structure; and

12 a command agent for executing an the selected executable feature
13 stored in association with a graphics element.

14 14. (Original) The system of claim 13 wherein the capture agent is
15 invoked by the application driver.

16 15. (Original) The system of claim 13 wherein the capture agent submits
17 the retrieved information to the application driver.

18 16. (Canceled)

19 20 17. (Currently Amended) The system of claim 13[6] wherein the application
21 driver deterministically selects one of the~~an~~ executable features that has not been
22 previously executed.

23 24 18. (Currently Amended) The system of claim 17 wherein the application
25 driver reviews an indicator to select the one~~an~~ executable feature not previously

1 executed.

2 19. (Currently Amended) The system of claim 13[6] wherein the
3 application driver deterministically selects executable features in a depth-first
4 mode of operation.

5
6 20. (Currently Amended) The system of claim 13[6] wherein the
7 application driver deterministically selects executable features in a breadth-
8 first mode of operation.

9
10 21. (Original) The system of claim 13 wherein the application driver
11 updates an indicator associated with the graphics element when an executable
12 feature stored in association with the graphics element is executed.

13
14
15
16
17
18
19
20
21
22
23
24
25

22. (Currently Amended) A method for systematically invoking an executable feature of a software application having a graphical user interface, the graphical user interface displaying a graphics element associated with the executable feature, the method comprising:

4 retrieving information descriptive of the at least one graphics elements
5 rendered during execution of the software application, the information including
6 an executable feature associated with each of the graphics elements;

7 storing an association between the each executable feature and the corresponding graphics element in a map data structure, wherein the map data
8 structure is accessible by an application driver for driving the software
9 application;

10 selecting one of the from the stored information an executable features that
11 has not been previously executed; and

12 executing the selected executable feature.

13
14 23. (Currently Amended) The method of claim 22 further comprising, in
15 response to executing the selected executable feature;

16 displaying exposing another second graphics element:

17 retrieving information descriptive of the othersecond graphics element,
18 the information including another second executable feature associated with
the othersecond graphics element;

19 storing another association in the map data structure, the other association
20 associating the othersecond executable feature in association with the
21 othersecond graphics element; and

22 selecting from the map data structure stored information a second the
23 other executable feature that has not been previously executed; and

24 executing the selected othersecond executable feature.

- 1 24. (Original) The method of claim 22 wherein the retrieving
2 comprises capturing information pertaining to the graphics element.
3
4 25. (Original) The method of claim 22 wherein the storing comprises updating
5 an indicator associated with the graphics element when an executable feature
6 stored in association with the graphics element is executed.
7
8 26. (Original) The method of claim 22 wherein the selecting comprises
9 reviewing an indicator to determine an executable feature not previously
10 executed.
11
12 27. (Original) The method of claim 22 wherein the selecting comprises
13 selecting executable features in a depth-first mode of operation.
14
15
16 28. (Original) The method of claim 22 wherein the selecting comprises selecting
17 executable features in a breadth-first mode of operation.
18
19
20 29. (Canceled)
21
22 30. (Canceled)
23
24 31. (Canceled)
25 32. (Canceled)
26
27 33. (Canceled)

1 34. (New) A computer-readable storage medium having computer-executable
2 instructions that when executed by a computer performs a method comprising:

3 a) determining a state of a target application based on a plurality of
4 graphics elements currently rendered on a display via a graphical user interface;

5 b) associating each of the graphics elements for the state with an
6 executable feature of the target application;

7 c) deterministically selecting one of the graphics elements that has not
8 been previously selected;

9 d) executing the executable feature associated with the selected graphics
10 element to obtain a new state; and

11 e) repeating steps a-d for the new state.

12 35. (New) The computer-readable storage medium of claim 34, further
13 comprising storing the association between each graphics element with its
14 corresponding executable feature in a map.

15 36. (New) The computer-readable storage medium of claim 35, further
16 comprising storing an indicator for each graphics element to indicate whether
17 the graphics element has been previously executed.

18 37. (New) The computer-readable storage medium of claim 35, wherein
19 deterministically selecting one of the graphics elements uses the map.

21 38. (New) The computer-readable storage medium of claim 34, wherein
22 deterministically selecting one of the plurality of graphics elements is
23 performed in a breadth-first manner.

24 39. (New) The computer-readable storage medium of claim 34, wherein

deterministically selecting one of the plurality of graphics elements is performed in a depth-first manner.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25